TC2100 RANDOLPH

COMMISSIONER FOR PATENTS
P.O. BOX 1450
ALEXANDRIA, VA 22313-1450
IF UNDELIVERABLE RETURN IN TEN DAYS

OFFICIAL BUSINESS

AN EQUAL OPPORTUNITY EMPLOYER

UNITED STATES POSTAGE
U.S. OFFICIAL MAIL
PENALTY FOR PRIVATE USE $300
PITNEY BOWES
02 1A
0004204479
$ 01.29⁰
SEP 27 2005
MAILED FROM ZIPCODE 22314

RECEIVED
OCT 0 5 2005
USPTO MAIL CENTER

IFW

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

32205 7590 09/27/2005

PATTI & BRILL
ONE NORTH LASALLE STREET
44TH FLOOR
CHICAGO, IL 60602

RECEIVED
OIPE/IAP
OCT 06 2005

EXAMINER	
JUNG, DAVID YIUK	
ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 09/27/2005

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/821,668	03/29/2001	Michael Dennis Ladwig	LIT-104/PRC-145	6164

TITLE OF INVENTION: METHOD AND APPARATUS FOR PROVIDING A SOFTWARE AGENT AT A DESTINATION HOST

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$1700	12/27/2005

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE REFLECTS A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE APPLIED IN THIS APPLICATION. THE PTOL-85B (OR AN EQUIVALENT) MUST BE RETURNED WITHIN THIS PERIOD EVEN IF NO FEE IS DUE OR THE APPLICATION WILL BE REGARDED AS ABANDONED.

HOW TO REPLY TO THIS NOTICE:

I. Review the SMALL ENTITY status shown above.

If the SMALL ENTITY is shown as YES, verify your current SMALL ENTITY status:

A. If the status is the same, pay the TOTAL FEE(S) DUE shown above.

B. If the status above is to be removed, check box 5b on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and twice the amount of the ISSUE FEE shown above, or

If the SMALL ENTITY is shown as NO:

A. Pay TOTAL FEE(S) DUE shown above, or

B. If applicant claimed SMALL ENTITY status before, or is now claiming SMALL ENTITY status, check box 5a on Part B - Fee(s) Transmittal and pay the PUBLICATION FEE (if required) and 1/2 the ISSUE FEE shown above.

II. PART B - FEE(S) TRANSMITTAL should be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). Even if the fee(s) have already been paid, Part B - Fee(s) Transmittal should be completed and returned. If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed and an extra copy of the form should be submitted.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Utility patents issuing on applications filed on or after Dec. 12, 1980 may require payment of maintenance fees. It is patentee's responsibility to ensure timely payment of maintenance fees when due.

PTOL-85 (Rev. 07/05) Approved for use through 04/30/2007.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), to: Mail **Mail Stop ISSUE FEE**
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

or Fax (571) 273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence including the Patent, advance orders and notification of maintenance fees will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications.

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

32205 7590 09/27/2005

PATTI & BRILL
ONE NORTH LASALLE STREET
44TH FLOOR
CHICAGO, IL 60602

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being facsimile transmitted to the USPTO (571) 273-2885, on the date indicated below.

(Depositor's name)

(Signature)

(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/821,668	03/29/2001	Michael Dennis Ladwig	LIT-104/PRC-145	6164

TITLE OF INVENTION: METHOD AND APPARATUS FOR PROVIDING A SOFTWARE AGENT AT A DESTINATION HOST

APPLN. TYPE	SMALL ENTITY	ISSUE FEE	PUBLICATION FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	NO	$1400	$300	$1700	12/27/2005

EXAMINER	ART UNIT	CLASS-SUBCLASS
JUNG, DAVID YIUK	2134	726-029000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list

(1) the names of up to 3 registered patent attorneys or agents OR, alternatively,

(2) the name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 ____________________
2 ____________________
3 ____________________

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document has been filed for recordation as set forth in 37 CFR 3.11. Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE

(B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☐ Corporation or other private group entity ☐ Government

4a. The following fee(s) are enclosed:

☐ Issue Fee

☐ Publication Fee (No small entity discount permitted)

☐ Advance Order - # of Copies ____________

4b. Payment of Fee(s):

☐ A check in the amount of the fee(s) is enclosed.

☐ Payment by credit card. Form PTO-2038 is attached.

☐ The Director is hereby authorized by charge the required fee(s), or credit any overpayment, to Deposit Account Number ____________ (enclose an extra copy of this form).

5. **Change in Entity Status** (from status indicated above)

☐ a. Applicant claims SMALL ENTITY status. See 37 CFR 1.27.

☐ b. Applicant is no longer claiming SMALL ENTITY status. See 37 CFR 1.27(g)(2).

The Director of the USPTO is requested to apply the Issue Fee and Publication Fee (if any) or to re-apply any previously paid issue fee to the application identified above.
NOTE: The Issue Fee and Publication Fee (if required) will not be accepted from anyone other than the applicant; a registered attorney or agent; or the assignee or other party in interest as shown by the records of the United States Patent and Trademark Office.

Authorized Signature ____________________ Date ____________

Typed or printed name ____________________ Registration No. ____________

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/821,668	03/29/2001	Michael Dennis Ladwig	LIT-104/PRC-145	6164

32205 7590 09/27/2005

PATTI & BRILL
ONE NORTH LASALLE STREET
44TH FLOOR
CHICAGO, IL 60602

EXAMINER	
JUNG, DAVID YIUK	

ART UNIT	PAPER NUMBER
2134	

DATE MAILED: 09/27/2005

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)

(application filed on or after May 29, 2000)

The Patent Term Adjustment to date is 847 day(s). If the issue fee is paid on the date that is three months after the mailing date of this notice and the patent issues on the Tuesday before the date that is 28 weeks (six and a half months) after the mailing date of this notice, the Patent Term Adjustment will be 847 day(s).

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Application Information Retrieval (PAIR) WEB site (http://pair.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571) 272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at (703) 305-8283.

Notice of Allowability	Application No.	Applicant(s)	
	09/821,668	LADWIG, MICHAEL DENNIS	
	Examiner	Art Unit	
	David Y. Jung	2134	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☒ This communication is responsive to 6/24/2005.
2. ☒ The allowed claim(s) is/are 1-41.
3. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
   a) ☐ All b) ☐ Some* c) ☐ None of the:
   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____.

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

4. ☐ A SUBSTITUTE OATH OR DECLARATION must be submitted. Note the attached EXAMINER'S AMENDMENT or NOTICE OF INFORMAL PATENT APPLICATION (PTO-152) which gives reason(s) why the oath or declaration is deficient.
5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   (a) ☐ including changes required by the Notice of Draftsperson's Patent Drawing Review (PTO-948) attached
   1) ☐ hereto or 2) ☐ to Paper No./Mail Date _____.
   (b) ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____.

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).
6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☒ Notice of References Cited (PTO-892)
2. ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
3. ☐ Information Disclosure Statements (PTO-1449 or PTO/SB/08), Paper No./Mail Date _____
4. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material
5. ☐ Notice of Informal Patent Application (PTO-152)
6. ☐ Interview Summary (PTO-413), Paper No./Mail Date _____ .
7. ☐ Examiner's Amendment/Comment
8. ☒ Examiner's Statement of Reasons for Allowance
9. ☐ Other _____.

DETAILED ACTION

Allowable Subject Matter

All claims are allowed. Claims 1-41 are allowed. The following is an examiner's statement of reasons for allowance: As noted in the amendment (such as at pages 10), the claimed inventions deal with the further code unit not being sourced by the originating host. This is not merely a matter of splitting the software agent, but also combining the data unit with a further code unit associated with the data unit, forming a destination agent. The prior art did not teach or suggest these features of the claims in the context of the other limitations of the claims.

Conclusion

Points of Contact

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Any response to this action should be mailed to:

Commissioner of Patents and Trademarks

Washington, D.C. 20231

or faxed to:

(571) 273-8300, (for formal communications intended for entry)

Or:

(571) 273-3836 (for informal or draft communications, please label "PROPOSED" or "DRAFT")

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David Jung whose telephone number is (571) 272-3836 or Greg Morse whose telephone number is (571) 272-3838.

David Jung

Patent Examiner

9/18/05

Notice of References Cited

Application/Control No. 09/821,668		Applicant(s)/Patent Under Reexamination LADWIG, MICHAEL DENNIS	
Examiner David Y. Jung		Art Unit 2134	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
	A	US-			
	B	US-			
	C	US-			
	D	US-			
	E	US-			
	F	US-			
	G	US-			
	H	US-			
	I	US-			
	J	US-			
	K	US-			
	L	US-			
	M	US-			

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Non-repudiation oblivious watermarking schema for secure digital video distribution; Zhou, W.; Rockwood, T.; Sagetong, P.; Multimedia Signal Processing, 2002 IEEE Workshop on 9-11 Dec. 2002 Page(s):343 - 346
	V	Punishing manipulation attacks in mobile agent systems; Esparza, O.; Soriano, M.; Munoz, J.L.; Forne, J.; Global Telecommunications Conference, 2004. GLOBECOM '04. IEEE Volume 4, 29 Nov.-3 Dec. 2004 Page(s):2235 - 2239 Vol.4
	W	A study on the system call for the protection of intellectual property rights on Linux base; Heun Kim; Dae-Joon Hwang; Dependable Computing, 2001. Proceedings. 2001 Pacific Rim International Symposium on 17-19 Dec. 2001 Page(s):295 - 298
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Non-repudiation Oblivious Watermarking Schema for Secure Digital Video Distribution

Wensheng Zhou, Troy Rockwood
Information Sciences Lab
HRL Laboratories, LLC.
Malibu, CA 90265, USA
Telephone: (310) 317-5278
Fax: (310) 317-5695
Email: (wzhou, rockwood)@wins.hrl.com

Phoom Sagetong
Integrated Media Systems Center
Dept. of Electrical Engineering-Systems
University of Southern California
Los Angeles, CA 90089-2564, USA
Telephone: (213) 740-0022
Email: sagetong@biron.usc.edu

Abstract—This paper presents a mechanism and algorithm for creating undeniable watermarks. It assumes a system where a content owner or provider uses outside agents to distribute its content. Content watermarked by distribution agents using this system will be undeniably recognizable by the content provider as originating with that distribution agent. That is to say that given N distribution agents, the content provider will be able to tell which distribution agent watermarked the content. The system does not allow any distribution agent to watermark content that would appear to have been watermarked by another agent and it does also not allow the content provider to watermark content that would appear to have been watermarked by a particular distribution agent. This allows the content provider to place a high degree of trust in the identification of the distribution agent and trace "leak" locations of pirated copies of videos.

I. Introduction

More and more digital multimedia data is distributed through public networks. Many approaches are available for protecting digital data; these include encryption, authentication, time stamping and watermarking. Most existing watermark schemes in distributed systems depend on a trusted third party (TTP) to verify the authentication of the watermark system. The secure delivery of images over open networks proposed by Augot et al [1] may encounter situations that a "trusted third party" can not be found that can be trusted by both parties. Other watermarking systems that we are aware of that aim to accomplish the same end goal must employ the services of a "trusted third party" to put watermarking keys in escrow to be presented upon demand if there is a dispute.

We propose a mechanism which does not need "trusted third party", every watermark is non-repudiation watermark, and can be used to identify the source of the watermark. Although there are many uses for this technology, the use for which it was developed is multimedia content distribution forensic analysis. In these cases, the multimedia content must be kept secret and not distributed by unauthorized agents. Should the content "leak" and become uncontrolled, it is desirable to locate the source of the leak so that appropriate action can be taken (punitive damages sought and security tightened for instance.) This multimedia watermarking mechanism allows content to be linked to the end user or distributor of the content, whichever is the responsible party in a way that corrective action (legal or technical) may be taken with confidence.

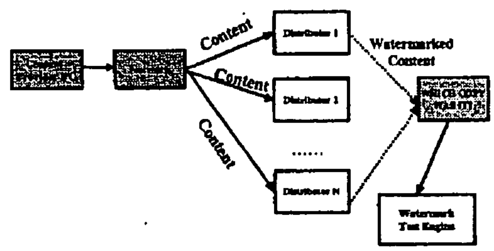


Fig. 1. Digital content distribution model with secure copy monitoring

II. Secure Digital Content Distribution Architecture

A. Distribution Model with Copy Leak Tracing

This paper introduces a novel multimedia watermarking mechanism that allows the non-repudiation of watermarked content. This is useful when the distribution of content needs to be known or proved. One such example is when copyrighted content and presentations of that content must be accounted for source and where the loss of control of that content could lead to monetary loss on the part of the content provider (who is assumed to also be the copyright owner). This non-repudiation watermark schema can be used for copy source tracking in a secure digital content distribution system which uses broadcasting technologies, such as satellite or multicast. The distribution model is shown in Figure 1. The content provider passes the valuable digital content to the content distributor to be distributed to all eligible clients, in this case, the clients are also distributors. All clients are required to put the watermarks into the digital content according to the proposed non-repudiation watermark schema so that content provider can trace the source of the leak if the copy is leaked. The novelty of the distribution model and non-repudiation watermark schema allows the reliable and non-repudiable watermark to fulfill the needs and trust

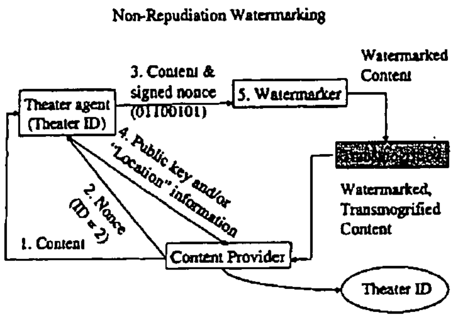


Fig. 2. Non-repudiate watermark scheme for digital multimedia distribution

of content provider and content consumers/presenters. This distribution model does not implement any protections to ensure that the watermark is applied properly, it is however assumed that both parties (the content provider and the distributor) willingly agree to follow the procedure as outlined. In the case where the distributor or the provider wishes to "cheat" the other by circumventing the watermarking procedure, other measures must be taken to ensure that this is not done [2]. Also, watermark attacks must also be addressed and considered in designing suitable watermark algorithm [3] for the system.

B. Non-repudiation Watermark Schema for Distribution

The non-repudiation watermark schema for digital multimedia distribution is depicted in Figure 2. This schema requires the use of public and private key encryption algorithms and assumes the participation of one content provider and at least one content distributor. Both the content provider and the distributor have their own private key that they do not share. This key is central to the identification of content watermarked by the distributors. First the content provider sends a file with the content to the distributor. This may be done in a variety of ways including but not limited to transmission of the content through a data network and distribution of the content on physical media (for instance CDROM's or DVD's). Once the content has been sent to the distributor, the distributor must contact the provider. This contact must be authenticated using "strong" authentication techniques. The exchange must be protected by "strong" encryption techniques. After authenticating with one another, the provider provides a random number to the distributor. The size of this random number is dictated by the watermark style to be used. Multiple watermark mechanisms may be used with this technology. The distributor generates a public and private key pair for the application of this watermark which we will call the watermark pair. The distributor uses the watermark pair private key to encrypt the random number passed to it by the provider. This encrypted number will be the watermark key. The distributor then watermarks the content with this key. After watermarking (or before, depending on the details of the watermark process), the distributor will transmit to the provider the public key of the watermark pair signed by its private key (not the watermark pair private key, the distributor private key)

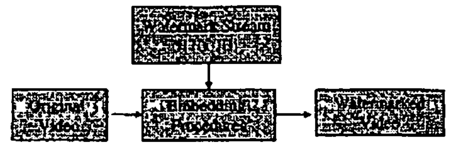


Fig. 3. Watermark embedding flow chart

along with information that will allow the provider to obtain the watermark key given the watermarked content. We will label this information "location information". However, it may not designate a location in the traditional sense of the word and will depend on the watermark technique selected. The provider archives this information so that the watermark may be detected later.

III. Non-repudiation Watermark Embedding and Detection Algorithms

To satisfy the non-repudiation watermark schema requirements, watermarking algorithms must have the following characteristics: First, the watermark should use a key from a large number space such that no two keys are likely to be identical if keys are chosen at random. Second, the watermark key can be detected given information other than the value of the key itself. Third, each copy has a unique watermark associated with a distinguished key for transaction information in digital content distribution systems. In our digital cinema scenario, we need at least of 56 bits of watermark payload to identify a movie at each specific theater at each show time. Fourth, it is nice to have blank detection or semi-blank detection watermark. The watermark detection agent should be able to detect out watermark without or with very limited information. Last and most importantly, we need to create non-fragile or robust watermarks. This is the most important requirement in this proposed watermark schema. To verify the watermark key, the watermark agent needs to reconstruct the encrypted key-stream exactly.

The above watermark requirements set up guidelines for the watermark algorithm design. In this paper, we develope an oblivious-detection watermark algorithm to allow the digital content provider automatically trace down the distribution source of the content without knowing the distribution agent's watermark key. Instead, the provider can extract the watermark key and verify irrefutably that the distributor generated it.

A. Watermark Embedding Algorithms

Within the non-repudiation watermark schema, the watermark embedding procedure is depicted in Figures 2 and 3. First, content provider will distribute the valued digital multimedia content to all its customers, such as theater agents. Then content provider will provide a unique nonce to each theater agent to uniquely identify the transaction. The nonce will be signed by the theater agent by using the theater's provider key to get its own unique watermark key. A watermarker is employed at each

theater agent's location as shown in Figure 2, where a watermark protected digital multimedia copy will be generated. The watermarker may use various watermark algorithms, however, they must satisfy certain specific requirements. To enforce the validation of this watermark schema, additional security measures must be used [2].

Our watermark algorithm [3] is robust to the attacks in both the temporal and spatial domains and to MPEG-recompression. We improve our previous *semi-oblivious* watermarking algorithm [3] to a fully *oblivious* watermark method. In our previous dynamic wavelet feature-based watermark algorithm [3], we cast watermark bits into the *energy* of the blocks of selected middle frequency subbands of each static wavelet-transformed frame. We modify an energy value of the selected block corresponding to the casting watermark bit. At the detector, we compare the energy of each block extracted from the watermarked (and attacked) video sequence with the corresponding original energy to determine the watermark bit. More detailed algorithm is described in [3]. Although the proposed semi-oblivious method fits in our proposed non-repudiation watermark schema, it requires the extra information to be transmitted for watermark detection. In this paper, while we will use the similar wavelet subbands to cast the watermark bits at the middle frequency subbands of static frames, we will introduce the prudent method to cast the watermark such that the detector will *not* need any information from the encoder, *an oblivious watermark detection*, and still remain robust to attacks.

After preprocessing the video as in [3], i.e. scene change detection, and temporal and spatial wavelet transformations the scene, we cast the watermark bits into the middle bands of the static frame of the video clip. Instead of separating the coefficients into multiple blocks as in [3], we introduce a novel methods of using polyphase transform for watermark casting. To simplify the explanation of the proposed algorithm, an example of the polyphase transform of one static frame of size 4×4 with 1-level wavelet spatial decomposition is illustrated in Figure 4. LL, LH, HL and HH subbands are composed of $\{X_1\ X_2\ X_5\ X_6\}$, $\{X_3\ X_4\ X_7\ X_8\}$, $\{X_9\ X_{10}\ X_{13}\ X_{14}\}$ and $\{X_{11}\ X_{12}\ X_{15}\ X_{16}\}$ respectively. We consider all pixels in LL subband as the roots of each tree. Each root has its children along all the other subbands. This is the famous zero-tree structure introduced previously in state-of-the-art wavelet image coders such as EZW [4] and SPIHT [5]. We then apply the polyphase transform to all the subbands. The polyphase transform will subsample the wavelet coefficients in row and column directions into multiple polyphase components. All components will eventually have similar characteristics to each other, e.g., the polyphase components will all have similar energy values.

Next, the nearby components are paired up. The two components that will be paired as a couple will have to be spatially located next to each other to make sure that the pair shares similar features. Here we choose components 1 and 2 to be the 1^{st} pair and components 3 and 4 to be the 2^{nd} pair. Only bit of watermark will be inserted into each polyphase-component pair. The rule to cast watermark bit "1" into the 1^{st} pair of components 1 and 2 is that the energy of the block which belongs to LH subband of component 1 (i.e. $E_1 = X_3^2$) must be *greater* than the energy of the block which belongs to the LH subband of component 2 (i.e. $E_2 = X_4^2$). In this work, we propose to modify X_3 and X_4 as shown in Equation (1) such that $E_1 > E_2$, where $i = 3$ and $j = 4$ and α is watermark strength where $0 \leq \alpha < 1$.

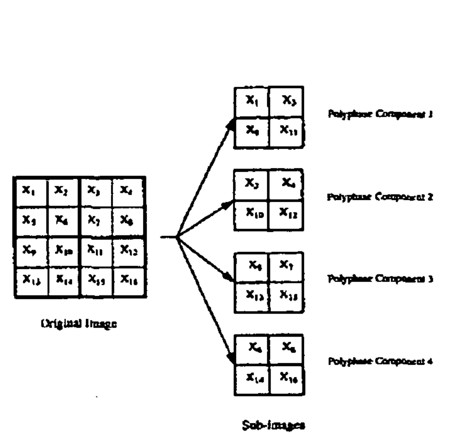


Fig. 4. An example of a polyphase transform occurs when a frame, which is assumed to have a size of 4×4, is segmented into 4 subbands of size 2×2 where X_i represents the i^{th} pixel, $i = 1, \ldots, 16$, from a frame of total 16 pixels. The image pixels can be grouped into successive polyphase components according to a particular spatial location in each original subband. Finally four polyphase components are generated.

$$X_i^w = (1+\alpha)\Big(\frac{X_i + X_j}{2}\Big)$$
$$X_j^w = (1-\alpha)\Big(\frac{X_i + X_j}{2}\Big) \quad (1)$$

To cast watermark bit "0", we perform the same process but, instead, setting $i = 4$ and $j = 3$. It is worth to note that the larger the α value, the larger the difference between X_i^w and X_j^w, the higher the robustness and the worse the perceptual quality will be. It is clear that α plays a main role in adjusting a tradeoff between robustness and perceptual quality of the watermarked data. We conduct the same process for HL subband (X_9 and X_{10}). Note that LL subband is not involved in watermark casting process because too much visual quality would be distorted if we cast watermark bits into it.

From equation (1), we first change the coefficient values of both components to the average value of both. The polyphase transform is chosen because the coefficients of each polyphase component from the same spatial location will have approximately the same value (i.e., $X_1 \approx X_3$). Therefore, changing them to the average value ($\frac{X_1+X_3}{2}$) will not dramatically affect their original values. However, it will equalize the amplitude of the coefficients before adjusting them based on the casting watermark bit. If there are other watermark bits, we will pursue inequality in this way on the watermark bits into next polyphase-components pair until we run out of watermark bits to cast. If there are some polyphase-components pairs left-over without

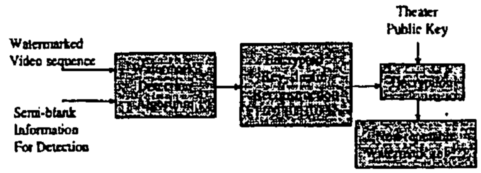


Fig. 5. Watermark detection flow chart

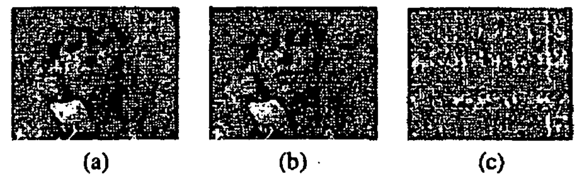


Fig. 6. Visual comparison between (a) original suzie frame and (b) watermarked suzie frame when (c) represents the watermark embedded

watermark casting, we can repeat the same watermark bits over to increase extra reliability. We pursue the same process for other static frames. To finalize the watermark casting process, we will reverse the process of reconstructing the movie into the raw data domain by doing an inverse spatial- and temporal-wavelet transformations as described in [3].

B. Watermark Detection Procedure

To detect the cast watermark pattern before performing decryption as shown in Figure 5, we repeat some of the processes in the watermark casting procedure described in [3], i.e., scene detection, temporal wavelet decomposition, and spatial wavelet decomposition. Next we compute the energies of two polyphase components in the pair of the selected subbands and make a comparison of the energies to each another. For example, in the system described in Section III-A, after we computed E_3 and E_4, if $E_3 > E_4$, we detect "1" as a watermark bit, otherwise watermark bit is "0". We perform the same process repeatedly until all the bits in every static frame have been detected. Furthermore, as described in [3], we may repeat the same process for every scene in the movie. We detect each watermark bit based upon the majority vote of watermark outcomes among all scenes. It is worth noting that, as described, our proposed watermark algorithm does *not* need use of the original movie or any information from the encoder to detect the watermark bit (oblivious watermarking). As described in Section II, This is very useful when a trusted third party and original copy do not exist in the security system environment.

After a candidate watermark key is extracted, iterative attempts are made to verify the distributor's identification by decrypting the key with the distributors' public key (provided at the watermarking time in schema of Figure 2) and then compared with the nonce in Figure 2 (given to the distributor at the start of the initial exchange.) If the decrypted watermark key matches the nonce, the content source has been successfully identified, if not, the producer goes on to the next distributor.

IV. Experimental Results

We simulated the watermark embedding and detection on the suzie and silent test sequences of 144 frames and assumed that each is one scene of a movie. Each frame has the size of 144×176 pixels. We then applied 3-level temporal wavelet decomposition and 4-level spatial wavelet decomposition to both sequences and end up with 18 static frames with 99 polyphase components each. We chose to cast the watermark to the finest HL and LH subbands and the second finest HH subband. In the digital cinema scenario, we need at least 56 bits of watermark payload to identify a movie at each specific theater at each show time. Therefore, we cast 10 bits in each static frame. Since all 56 bits can be cast in 6 static frames, the other 12 static frames are cast with the same watermark to increase the robustness of the watermark. We first cast 60 watermark bits to the suzie and silent sequences with watermark strength $\alpha = 0.1$. The PSNR of the watermarked sequences were at 44.23 dB and 41.96dB, respectively. Figure 6 shows the visual differences between the original and watermarked images, and no visual difference of the two images can be noticed.

We then applied an MPEG compression attack to the watermarked "silent" sequence changing bit rates. All embedded watermark bits were correctly detected up to compression ratio of 1/14 of the raw data. For the temporal attacks, we can correctly detect all watermark bits when we subsampled frames from 25 fps to 12.5 fps and when we cropped the frames, making the sequence shorter by deleting the frames, from the beginning/end up to 66.7% of the total number of frames. To detect a watermark after frame subsampling/dropping attacks, we substituted the missed frames with the average of the frames available. This means our proposed algorithm can also tolerate the frame-averaging attack. Finally, we tested the proposed watermark with spatial attacks. Watermark survives both when rows and columns were subsampled by 2, i.e., only 1/4 of the original size is retained. We tested cropping the rows on the top and bottom and the columns on the left and the right of each frame. Image size-cropping can be used in real applications, such as converting a wide-screen movie into a normal TV screen-size movie. Our watermark survived very well when up to 30.68% of size of the image was cropped.

References

[1] D. Augot, J-M. Boucqueau, J.F. Delaigle, C. Fontaine, and E. Goray, "Secure delivery of images over open networks," *Proceedings of the IEEE, Vol. 87, No. 7*, July 1999.

[2] T. Rockwood, W. Zhou, B. Ryu, and Y. Zhang, "Secure systems and methods for digital cinema distribution," *Techinical Report, Information Sciences Lab, HRL Laboratories, LLC.*, June 2001.

[3] P. Sagetong and W. Zhou, "Dynamic wavelet feature-based watermarking for copyright tracking in digital movie ditribution systems," *In 2002 IEEE International Conference on Image Processing (ICIP), Rochester, New York*, Sept. 2002.

[4] J. M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Trans. on Signal Processing, Special Issue on wavelets and Signal Processing, 41(12): pp. 3445-3462*, Dec 1993.

[5] A. Said and W. A. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. on Circuits and Systems for Video Technology, vol. 6, no. 4, pp. 243-250*, June 1996.

Punishing manipulation attacks in mobile agent systems

Oscar Esparza Miguel Soriano Jose L. Muñoz Jordi Forné
Technical University of Catalonia
Jordi Girona 1 i 3.
08034 Barcelona, Spain.
Tel. +34 934010972 Fax. +34 934011058
{oscar.esparza,soriano,jose.munoz,jforne}@entel.upc.es

Index Terms—**security, mobile agent, malicious host, software watermarking, host revocation.**

Abstract—**Mobile agents are software entities consisting of code, data and state that can migrate autonomously from host to host performing some actions on behalf of a user. Unfortunately, security issues restrict the use of mobile agents despite its benefits. The protection of mobile agents against the attacks of malicious hosts is considered the most difficult security problem to solve in mobile agent systems. In a previous work, the Mobile Agent Watermarking approach (MAW) was presented as a new attack detection technique to aid to solve the problem of the malicious hosts. This approach was based on embedding a fixed watermark into the mobile agent. In this paper, some improvements are introduced to MAW. Instead of a fixed watermark, the origin host embeds a watermark that can change dynamically during execution. In each host, the marked code creates a data container where the watermark will be transferred and the results will be hidden. When the agent returns home, the origin host verifies the execution integrity by applying a set of integrity rules to the containers. This paper also explains how MAW can be used to punish the malicious hosts by using a Trusted Third Party, the Host Revocation Authority.**

I. Introduction

Mobile agents are software entities consisting of code, data and state and that can migrate from host to host performing some actions autonomously on behalf of a user. The use of mobile agents saves bandwidth and permits an off-line and autonomous execution in comparison with habitual distributed systems based on message passing. For this reason, mobile agents are especially useful to perform functions automatically in almost all electronic services, like e-commerce, data mining or network management. Despite their benefits, massive use of mobile agents is restricted by security issues.

This paper introduces some improvements to the Mobile Agent Watermarking approach (MAW) [6]. Instead of a fixed watermark that is located into the results in some positions previously known by the agent sender (or origin host, as it is also called), the watermark can change dynamically during execution. Before sending the agent, the origin host embeds a watermark into the agent's code by using software watermarking techniques [3]. During the execution in each host, the agent creates a data container that will be used later to verify the execution integrity and to hide the results. The agent transfers the watermark to the container by putting any kind of avaliable data inside of it in an ordered way. When the execution finishes, the results are also fitted into the container. When the agent returns to the origin host, it applies a set of integrity rules to all the data containers. These rules can be inferred from the modifications performed in the agent's code during the watermark embedding. If a container does not fulfill the rules, this means that the corresponding host is malicious. The proposal not only detects manipulation attacks performed during the agent's execution, but it also proves the malicious behavior of the host. This paper introduces how MAW can be used to punish the malicious host by using a Host Revocation Authority (HoRA from here on) [8]. The HoRA must be considered an independent Trusted Third Party (TTP) in a mobile agent system, like the Certification Authority is considered in the Public Key Infrastructure (PKI). The HoRA stores a database with some information about the revoked host, i.e. those hosts that have been proven malicious. Before sending an agent, each origin host consults the revocation information in order to delete all the revoked hosts from the agent's itinerary. As a result, the revoked hosts will not execute agents any more.

The rest of the paper is organized as follows: Section II describes the existing approaches to protect mobile agents; Section III details the improvements introduced in MAW; Section IV explains how to punish the malicious hosts by using MAW and the HoRA. Finally, some conclusions can be found in Section V.

II. Malicious hosts

The attacks performed by a malicious host that is executing the mobile agent are considered, by far, the most difficult problem to solve regarding mobile agent security. Notice that while it is possible to assure the integrity and authentication of the code, the data or the results that come from other hosts by using digital signature or encryption techniques, it is difficult to detect or prevent the attacks performed by a malicious host during the agent's execution. Malicious hosts could try to get some profit of the agent reading or modifying the code, the data, the itinerary, the communications or even the results due to their complete control on the execution. The agent cannot hold a decryption key because the hosts could read it. Furthermore, it is not sure that the host runs the complete code in a correct manner, or it simply does not allow the migration.

There are two main types of protection techniques: (1) attack avoidance approaches, that try to avoid the attacks before they happen; and (2) attack detection approaches, whose aim is detection after the attack has been performed. Detection techniques are less effective for services where benefits for tampering a mobile agent could be greater than the possible punishment. In those cases, attack avoidance approaches are recommended. Unfortunately, there is no current approach that avoids attacks completely.

A. Attack avoidance approaches

The simplest solution to avoid attacks is sending the agent only to trusted hosts, i.e. hosts that are not expected to attack the agent [11]. Obviously, this proposal is not useful in an open network like Internet because there are few trusted hosts. Yee introduces the idea of a closed tamper-proof hardware subsystem [15] where agents can be executed in a secure way, but this forces each host to buy a hardware equipment and to consider the hardware provider as trusted. Roth presents the idea of cooperative agents [12] that share secrets and decisions and have a disjunct itinerary. This fact makes collusion attacks difficult, but not impossible. Hohl presents obfuscation [9] as a mechanism to assure the execution integrity during a period of time, but this time depends on the capacity of analyzing the code of the malicious host. The use of encrypted programs [13] is proposed as the only way to give privacy and integrity to mobile code. The executing hosts run the encrypted code directly, and hence a decryption function is needed to recover the results. In [2] the approach is improved in the way that agents can traverse multiple hosts. In [1] the scheme allows the agents to take decisions while traveling by using a TTP. The difficulty here is to find functions that have the necessary properties, i.e. functions that can be executed in an encrypted way. Lately, in [5] a secure privacy homomorphism was presented, but its use is still limited to perform functions of an arithmetical nature.

B. Attack detection approaches

In [10], the authors introduce the idea of replication and voting. In each stage, a set of hosts execute the agent in a parallel way and send several replicas of the agent to the next stage. This offers a fault-tolerant mechanism to execute agents, but only can be used as an attack detection approach in those scenarios in which the hosts in the same stage are independent, i.e. they must have different interests to attack an agent. In [14], Vigna introduces the idea of cryptographic traces. During execution, the agent takes traces of instructions that alter the agent's state due to external variables. The origin host will only ask for the traces if it suspects that an executing host acted maliciously. Despite being the most widely known attack detection approach, it still has some major drawbacks that deter its implementation. At first, the executing hosts must store the traces for an indefinite period of time because the origin host can ask for them. Furthermore, verification is performed only in case of suspicion, but how a host becomes suspicious is not explained. Despite some of the drawbacks of this proposal were solved in [7], its use is still limited.

III. Improving the Mobile Agent Watermarking Approach

In [6], the authors introduced the Mobile Agent Watermarking approach, a lightweight attack detection approach that permits to verify the execution integrity without thinking in terms of suspicion. The origin host embeds a fixed watermark into the code in order to detect manipulation attacks. As a result, the running of the agent creates marked results. When the agent returns, the origin host looks for the watermark into the results. If a watermark differs from the expected one, this means that the corresponding executing host is malicious.

This paper introduces some improvements to MAW to make the proposal more flexible and secure. Before sending the agent, the origin host embeds a watermark into the agent's code. In each host, the running of the agent creates a data container where the watermark will be transferred. The agent can put any kind of available data into the container, for example dummy data, input data or even intermediate variable values. When the execution finishes, the results are also fitted into the container. Consequently, the transferred watermark changes dynamically during the execution. When the agent returns to the origin host, it applies a set of integrity rules to all the data containers. These rules must be inferred from the modifications performed in the agent's code during the watermark embedding. If a container does not fulfill the rules, this means that the corresponding host is malicious.

The rest of the section explains how the watermarks can be embedded into the mobile agent and how this proposal can be used to detect manipulation attacks.

A. Watermark embedding and transference

Before sending the agent, the origin host embeds a watermark into the agent's code by using software watermarking techniques [3]. These techniques are not used to protect the agent's copyright, but for detecting manipulations performed during execution. In each host, the execution of the marked code creates a logically-structured data container where the watermark will be transferred. During execution, the agent can put any kind of available information into the container, for example dummy data, input data, intermediate variable values or data that come from communications. Finally, the results are also fitted into the container. In fact, the data of the container can be made up of:

- Fixed values located in some positions previously known by the origin host, like it was in [6].
- Values that can change dynamically during the execution and fulfill a set of logical rules.

During the transference process, the agent diffuse (repeat values) and confuse (change values) all this information into the container. For this reason, the way this information is put into the container and the information itself constitute the transferred watermark. In short, the container is the digital cover where the agent's code must transfer the embedded watermark, and hence it can be used as a proof to verify the execution integrity. This transference process is shown in Figure 1. Furthermore, the container is hiding the results from malicious hosts, that is to say, a malicious host should

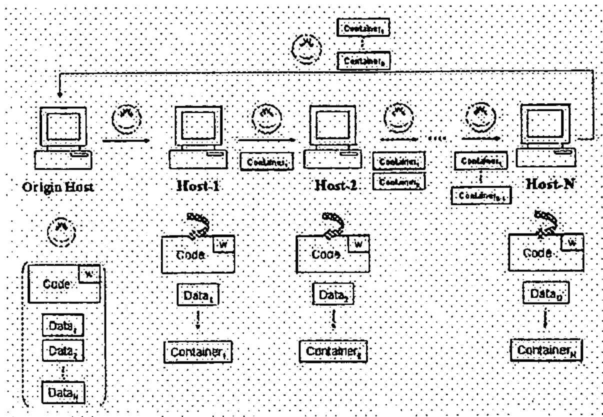


Fig. 1. Watermark transference during the agent's execution

not tell the difference between the results and the transferred watermark.

B. *Detecting manipulation attacks*

When the agent returns home, the origin host tries to detect the attacks performed during execution. To do so, the origin host verifies that all the containers fulfill a set of integrity rules. These rules can be inferred from the modifications performed over the original agent's code to embed the watermark. If a container does not fulfill the rules, this means that the corresponding host modified the mobile agent, so it is malicious. Notice that the way the origin host uses to verify the execution integrity is the same for all the hosts, but this does not mean that all the containers have the same watermark. In fact, the transferred watermark is different because it depends on the execution.

The alteration of the watermark not only detects manipulation attacks, but also proves the malicious behavior of the host. In Section IV is explained how the tampered containers can be used to punish the malicious hosts by using a TTP, the Host Revocation Authority [8].

IV. Punishing Manipulation Attacks

Attack detection approaches are not enough to protect the agent on their own. This kind of mechanisms must be attached with some punishment policies. Usually, a host will turn into malicious behavior only in case that the benefits for tampering the agent would be greater than the punishment. Thus, the harder the punishment, the less attacks will be performed by the hosts. This paper introduces how the mobile agent watermarking approach can be used to punish malicious hosts.

A. *Punishment policies*

Little attention has been paid to punishment mechanisms in the literature. This section summarizes some of the conducts that an origin host can follow when it detects an attack:

- The origin host does not take into account the partial results[1] of the executing hosts detected as malicious. These results are compromised, so they can be discarded automatically. On the contrary, if the results depend on the execution of more than one host, the whole execution is compromised and the origin host can only send the agent again removing the malicious hosts from the itinerary. This is the easiest punishment, as it only depends on the origin host, but it is also the weakest. For instance, there are mobile agent scenarios where the agent does not take results of execution, so there is nothing to discard and hence there is no real punishment to the malicious host. Furthermore, the detected malicious hosts can continue attacking other agents.
- The origin host creates a blacklist that contains all the executing hosts that attacked its agents. The origin host will not send agents to them any more. For instance, in an e-commerce scenario the origin host will not buy more products to a server that has been detected as malicious. This is an improvement of the previous behavior because a malicious host can only attack an origin host once. This punishment is easy to implement because it only depends on the origin host. However, a malicious host can continue attacking several different origin hosts.
- A group of origin hosts shares a common blacklist that contains all the malicious hosts. A host is introduced into this common blacklist if it is detected attacking an agent owned by an origin host of the group. All the malicious hosts of the common blacklist will not receive mobile agents from this group of origin hosts any more. However, a problem arises when there are origin hosts that do not trust each other. For instance, a malicious origin host can adversely affect an honest executing host by including it into the common blacklist. Consequently, no mobile agents will be sent to this honest host.
- A TTP stores and manages the common blacklist with all the hosts that acted maliciously. Obviously, attack detection is not enough, but also proving of the malicious behavior before the TTP adds a new host to the blacklist. In [8], the authors introduce a new entity in the mobile agent system to solve the lack of an entity with punishment capabilities. The Host Revocation Authority (HoRA) must be considered an independent TTP in a mobile agent system, like the Certification Authority is considered in the PKI. The HoRA stores a database with some information about the revoked host, i.e. those hosts that have been proven malicious. Before sending an agent, each origin host consults the revocation information in order to delete all the revoked hosts from the agent's itinerary. The real strength of this punishment mechanism lies in dissuading the executing host from being malicious because they can be revoked and consequently they will not receive mobile agents any more.

B. *Punishing attacks with MAW*

This paper presents how to use jointly MAW as the attack detection mechanism and the HoRA as the TTP in charge of the punishment. If the origin host detects that a container has been tampered, it can start a revocation protocol in order to

[1]Partial results are those that depend on the execution performed only in one host.

add the malicious host to the internal database of the HoRA. Before explaining this revocation protocol, some notation used in the message and agent passing must be introduced:

- We denote a mobile agent that moves from host x to host y as $Agent_{x \to y}()$.
- We denote a message from host x to host y as $Message_{x \to y}()$.
- We denote the signed copy of document D as $sign_{\alpha}[D]$, where α is the signing host identifier.
- We denote the One-Way Hash Function value (hash from here on) of document D as $H(D)$.

For easiness reasons, a single-hop scenario has been used for the explanation. These are the steps that the principals must follow:

1) Firstly, the origin host sends the agent to perform some actions on behalf of the user:

$$Agent_{O \to 1}(A)$$

where $A = sign_O[Code, Data, H(Rules)]$. The agent carries the code, some data and the hash of the rules in order to link the rules to this execution.

2) The executing host receives the agent and extracts the code and the data. Now consider that this host acts maliciously modifying the agent instead of executing the code directly. So then, the execution will create a tampered container. The host sends the following agent to home:

$$Agent_{1 \to O}(B)$$

where $B = sign_1[A, Container_1]$. B must contain A in order to link the code, the data and the rules with the container.

When the agent arrives to the origin host, it applies the integrity rules to the container. As the host modified the execution, the container $Container_1$ will not fulfill the integrity rules because the watermark has been modified. Consequently, the origin host starts a revocation process:

3) The origin host sends all the proofs of the execution to the HoRA:

$$Message_{O \to HoRA}(sign_o[B, Rules]).$$

The HoRA receives the revocation query and starts checking the proofs. As all the messages (including the agent) are properly signed, none of the entities can perform a repudiation attack. Firstly, the HoRA verifies that the $Rules$ match with the hash value $H(Rules)$ to verify that the code, the data, the container and the rules come from the same execution. Next, the HoRA must verify the execution integrity, but it cannot execute the agent exactly in the same way than the executing host because the container does not have all the input data (if so, this solution would be equivalent to the cryptographic traces approach [14]). The way to verify that the execution has not been tampered is by applying the integrity rules to the containers. The problem arises because these rules are not public, only the origin host knows them. So then, the HoRA needs proofs that these rules match the agent's code. This can be done by executing the agent (once or several times) with random input data. As the integrity rules have been inferred directly from the marked code, any container created with this code will fulfill the rules, independently from the input data. The integrity rules can be considered valid if the new container created during this random execution fulfills the rules[2]. Finally, the HoRA can verify if the executing host acted maliciously by applying the integrity rules to the container. The host will be revoked in case its container does not fulfill the rules.

4) Finally, the HoRA sends a message to the origin host with the revocation result:

$$Message_{HoRA \to O}(sign_{HoRA}[Revoked, Tampered\ execution]).$$

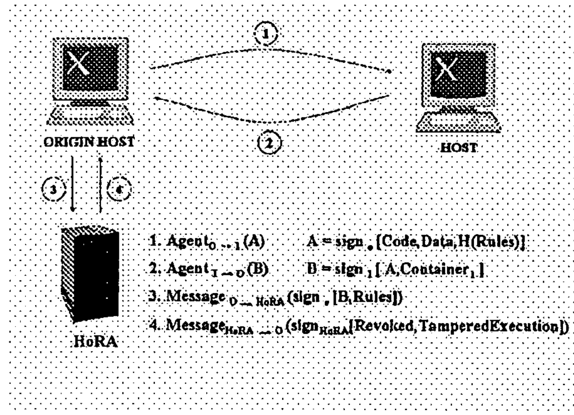


Fig. 2. Malicious host revocation using MAW

Figure 2 shows the complete process in a graphical way.

C. Advantages

MAW is a lightweight attack detection approach if it is compared to the most widely known, the cryptographic traces approach [14]. These are some of the advantages of MAW regarding the use of traces:

- The size of the containers is determined by the programmer and can be little enough to make the agent carrying them. On the contrary, the traces are not sent to the origin host with the agent because their size depends on the amount of input data, that can be huge.
- The origin host can verify the execution integrity of all the hosts because the containers return with the agent. On the other hand, in the cryptographic traces approach the verification is performed in case of suspicion because the origin host does not have the traces of all the hosts.
- In MAW, the executing hosts do not need to store any kind of proof. On the contrary, the hosts must store the traces for an indefinite period of time because the origin host can ask for them in case of suspicion.

[2] If the random containers do not fulfill the integrity rules, this means that the origin host is acting dishonestly trying to revoke an honest host. As a result, the HoRA could take a disciplinary measure to the origin host.

 0-7803-8794-5/04/$20.00 © 2004 IEEE

- In MAW, the origin host has to apply the rules to the containers to verify the execution integrity. In the cryptographic traces approach, the origin host must ask for the traces of the suspicious hosts to execute the agent again.
- To verify the execution integrity, the HoRA needs to execute the agent again with random input data to validate the rules, and later apply the rules to the containers. In the cryptographic traces approach, the origin host must execute the agent again with the input data of the traces. This means that both approaches have a similar cost to the HoRA.

D. Drawbacks

These are the main drawbacks that can be found in the Mobile Agent Watermarking approach:

- The origin host must embed the watermark into the agent's code and must infer the rules from these modifications.
- There is an increase in the original code size. Embedding a watermark always means that some overhead is added to the code.
- The mobile agent must carry a data container for each host, instead of just the results of execution.

E. Attacks

These are the main attacks that the malicious hosts can perform to MAW:

- Eavesdropping: all non-encrypted data in the agent can be read by the hosts. Although it is possible to modify the agent to make it harder to analyze, for instance by using obfuscation [4], a malicious host with enough time can guess the intentions of the agent.
- Manipulation: a malicious host can manipulate any part of the agent (the code, the data, the execution flow, the communications and even the results) to achieve an execution on its own profit. The aim of the malicious hosts is modifying the agent without altering the watermark, because any change in the watermark can be used as a proof to punish them. In this sense, the strength of MAW is making the watermark imperceptible enough to an observer, because a malicious host will be easily detected if it tries to modify the agent with no knowledge about how these modifications will alter the container.
- Collusion: it is difficult that a group of colluding hosts guesses the transferred watermark by comparing their containers, because the watermark is different (dynamically generated) for each host.

V. Conclusions

In this paper, some improvements have been added to the MAW approach. Instead of a fixed watermark like it was in [6], the origin host embeds a watermark that changes dynamically during execution. This makes the proposal more secure and flexible. In each host, the agent's code creates a container to transfer the watermark of the code and to hide the results. These containers are the proof of the good or bad behavior of the hosts. When the agent returns home, the origin host applies a set of integrity rules that the containers must fulfill. These integrity rules can be inferred from the modifications performed in the agent's code during the watermark embedding. If a container does not fulfill the rules, this means that the host has modified the agent during execution.

Additionally, this paper presents how to use jointly MAW and the HoRA to verify the execution integrity and to punish the malicious hosts. When the origin host detects a manipulation attack, it sends the proofs of the malicious behavior to the HoRA. If finally the HoRA considers that the proofs are valid, the new malicious hosts is included into its internal database. As a result, this malicious host will not receive agents any more.

Acknowledgments

This work is supported by the Spanish Research Council under the project DISQET CICYT TIC2002-00818 and the European Research Council under the project UBISEC (IST-FP6 506926).

References

[1] J. Algesheimer, C. Cachin, J. Camenisch, and G. Karjoth. Cryptographic security for mobile code. In *IEEE Symposium on Security and Privacy*, 2001.

[2] C. Cachin, J. Camenisch, J. Kilian, and Joy Müller. One-round secure computation and secure autonomous mobile agents. In *27th International Colloquium on Automata, Languages and Programming (ICALP)*, volume 1853 of *LNCS*. Springer-Verlag, 2000.

[3] C. Collberg and C. Thomborson. Software watermarking: Models and dynamic embeddings. In *Principles of Programming Languages 1999, POPL'99*, 1999.

[4] C. Collberg, C. Thomborson, and D. Low. A taxonomy of obfuscating transformations. Technical Report 148, The University of Auckland, 1997.

[5] J. Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism. In *5th Information Security Conference (ISC 2002)*, volume 2433 of *LNCS*. Springer-Verlag, 2002.

[6] O. Esparza, M. Fernandez, M. Soriano, J.L. Muñoz, and J. Forné. Mobile agent watermarking and fingerprinting: tracing malicious hosts. In *Database and Expert Systems Applications (DEXA 2003)*, volume 2736 of *LNCS*. Springer-Verlag, 2003.

[7] O. Esparza, M. Soriano, J.L. Muñoz, and J. Forné. A protocol for detecting malicious hosts based on limiting the execution time of mobile agents. In *IEEE Symposium on Computers and Communications - ISCC'2003*, 2003.

[8] O. Esparza, M. Soriano, J.L. Muñoz, and J. Forné. Host Revocation Authority: a Way of Protecting Mobile Agents from Malicious Hosts. In *International Conference on Web Engineering (ICWE 2003)*, volume 2722 of *LNCS*. Springer-Verlag, 2003.

[9] F. Hohl. Time Limited Blackbox Security: Protecting Mobile Agents From Malicious Hosts. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[10] Y. Minsky, R. van Renesse, F. Schneider, and S.D. Stoller. Cryptographic Support for Fault-Tolerant Distributed Computing. In *Seventh ACM SIGOPS European Workshop*, 1996.

[11] J. Ordille. When agents roam, who can you trust? Technical report, Computing Science Research Center, Bell Labs, 1996.

[12] V. Roth. Mutual protection of cooperating agents. In *Secure Internet Programming: Security Issues for Mobile and Distributed Objects*, volume 1906 of *LNCS*. Springer-Verlag, 1999.

[13] T. Sander and C.F. Tschudin. Protecting mobile agents against malicious hosts. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[14] G. Vigna. Cryptographic traces for mobile agents. In *Mobile Agents and Security*, volume 1419 of *LNCS*. Springer-Verlag, 1998.

[15] B.S. Yee. A sanctuary for mobile agents. In *DARPA workshop on foundations for secure mobile code*, 1997.

A Study on the System Call for the Protection of Intellectual Property Rights on Linux Base

Heun Kim, Dae-Joon Hwang
The School of Electrical and Computer Science, Sung Kyun Kwan University,
300 Chunchun-Dong, Jangan-Ku, Suwon 440-746, South Korea
heunyong@mail.skku.ac.kr

Abstract

The 21st century we currently live in is the age based on knowledge and information. It is a computer that provides power, which can lead the new wave of information. The digitalization of information is expressed as the digital revolution. Recently, the more important the protection of Intellectual Property Rights(IPR). The variety methods for protection of intellectual property rights are encryption and digital watermarking and access control and so on. In this paper, we propose RPLA system of active tracking mechanism for IPR protection. Also, protective model of intellectual property can use in a protective work of digital contents, adaptive agent and system level protection on client machine.

In conclusion, we describe the implementation of our model in linux system.

Key words : IPR, systemcall, hooking, kernel,RPLA

1. Introduction

In the past, the most important factors have been that of a visible capital and material. However, it has been changing gradually to an invisible knowledge and information from the past.

Linux is easy to modify, and it has been praised for its stability through the verified running systems by many users. It runs on the whole PC, and spread out dramatically because of the low cost installation. In terms of the current trend, it is easy to forecast hereafter that Linux will be the most important system in the major server market. Now is the era of Internet. we need to put on illegal software reproduction on cyberspace, becoming a serious social issue as enlarging Internet market.

As above, these actions must be prohibited. It is natural to make a new rule, to open information to the public, to authorize the right and to protect the creator who provides information.

In this paper, it will introduce our suggested model, which is differentiated with the existing intellectual property technology, and to provide explanation about that model.

2. Construction and function of RPLA System

Either online or offline, RPLA system is able to protect and track of many sources which are used in computer, prepared for digital distribution of digital contents.

Those protect able sources are passive resources (file, directory, port and so on) and active resources (process, thread) and those source are contain in adaptive agent technology and block of illegal approach and use.

2.1. The total structure of the suggested model.

1) Roles of the sever program
- The basic role of this is giving lots of missions to the agent.
- Receiving and collecting the results from the agents who are dispatched to several computers.
- It can get various statistics and gives new missions that are based on the reported results.
- Supply Network tool or GUI circumstance for convenience of manager.

2) Roles of the agent.
- Basically, Agents will be dispatched to computers or sites, which are necessary to monitor.
- Dispatched agents carry out their missions that is given by severs. Mission means duty and order which has to be protected and monitored for the resources of clients.
- The agents save the result to server when it is on-line and save to any place when it is off-line.

2.1.1 Mission Control

Mission control plays roles that decide the specific works and orders to be carried out by the agent, and transfer these. The order and work is completed one more union. Practical or impractical of the order will be decided through condition of AND, OR, according to demand of protected clients among the missions. The mission which is executed by the agent, will be all changed according as the agent dispatch to which IP area or which user's computer. The kinds of resources that will be monitored by the agent are divided into the active and passive element as shown in figure 3-1. Digital content, which is a main target to monitor, is classified into File, so this comes under the active resource. Identification is to

distinguish the site and target that the agent will dispatch to and work. Number of count is part that decides the number of using and the right of user will be altered depending on who users are. Group resource is the part of managing that means the resources, which the agent will monitor, and the user who will use the resource to be banded one unit as a group.

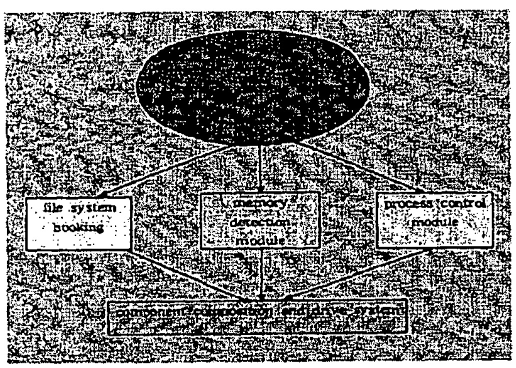
As above, the mission control is that various working and orders which showed in it, are given by consisting of agents which is to suit every computers.

2.1.2 The adaptive agent

The adaptive agent is operating a proper monitor module component to watch and to preserve the resource from the computer, which is dispatched agents, according to the site and an IP address. The agent has to operate downloaded the working component, in the case of receiving the duty for extended works and orders, because the agent is loading the minimum component. And also the adaptive agent technology implies making the appropriate agent to preserve and watch of specific sites and computers.

2.1.3 System hooking

The almost digital contents are file. Such as picture, text, and multimedia, these are all consisted of file. File hooking system enables file to protect and control itself in every hour. The model 2-1 is about RPLA construction module that consisting of file system hooking and the overall contents about file system hooking is as following in chapter 4.



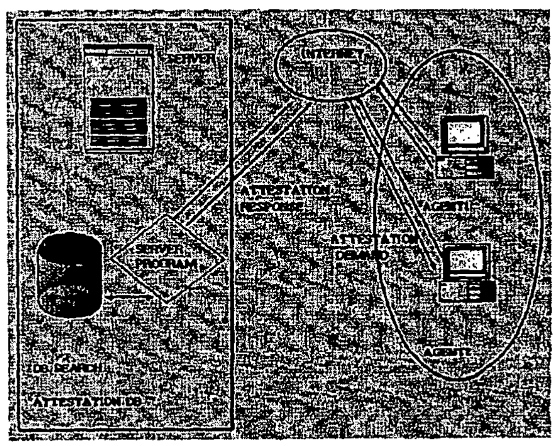
[The figure 2-1. RPLA construction module]

2.1.4 User authorization

This means series of process what the agent distinguish from who is a proper user when the user try to use a resource. The agent has now watched this resource. There are three ways to authorize user from the agent of the suggested model. The standard of classifying these is different depends on the computer is on-line or off-line, and depends on whether the resources which is necessary to keep watch, are many or small. When exchange the data connecting with server in the on-line, use TCP protocol. Firstly, when the computer which is dispatched the agent, is an off-line, it cannot get any related information with the authorization because it cannot connect with sever programs. Therefore, the agent has to have information, which relates with the information of the resources that will watch basically, the information of user who can use those resources, and the right of using number. Secondly, when the number of the resources are a lot, even the computer is an on-line, the agent has to find out the other way if the area is to big to watch.

The ways are as figure 3-6. In the case of used the resources, which is on monitor, the agent requests for an authorization to the server program and that program reply to the requests to the agent, in order to authorize of the resources.

This method minimizes the agent's burden of information storage but it might put stress on network. It may causes problem when the network has an obstacle.



[The figure 2-2. User attestation method for RPLA]

Thirdly, it is a case that number of the protective resources is not that many even though it is on-line. At that time, basically it loads the other information into the agent rather than the protective resources information and a user's information of the resource. It also stores into the agent's database from the server program by downloading the new added resourced information or user's information.

When that happens, it is faster than getting an authorize from connecting by sever each time, and it can be prepared for network area or it's error just like the second case stated above.

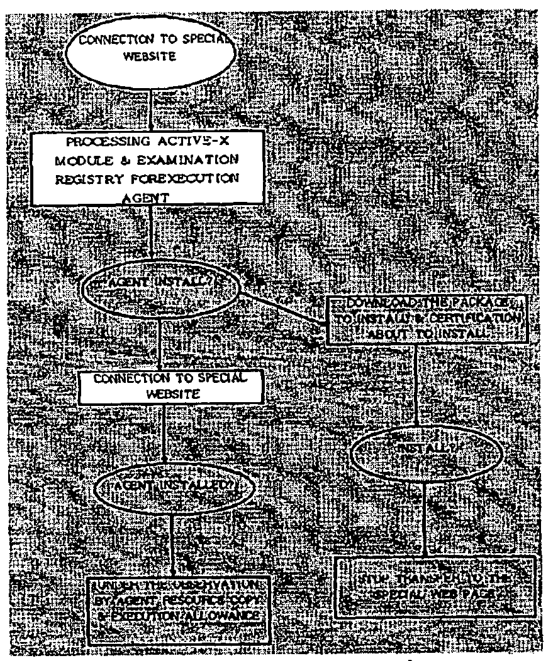
3. How to dispatch the agent.

This method means how to install and run the agent program into the computer system, which wanted to be monitored.

The first, when the user runs portable disk (CD-ROM,

floppy disks) in a computer, the computer installs the program automatically. The agent runs it every time when the computer is booting by user and the user should not be able to notice it is running. In order to hide from the users, the function called stealth is used. If it is used, the agent's running stage does not appear in the process list. As explain about the realization of the stealth's function, it is to resister into service module from running module.

The second, this is a way to dispatch the agent while it is on-line same as the user wants to be downloaded file which is linked by a special web site. Refer to the figure 3. An active component, which is embedded in a web site, runs when it is contacted by specific web site. And it tests out whether the agent's module is running in a connector's computer or not. If not, the package that installs the agent module is downloaded and it runs to install it. If the user avoids installing of it, they cannot open that site. Therefore, the user must install the agent module in that site.



[The figure 3 Agent installation process]

3.1 appropriate manners for decision of the resource's right and an illegal use of the specific resources.

To use easily the protective resources and the management of the user's information who will use that resources. Under a group, many resources and many user's information are registered. The user who is registered in that group is able to use the registered resources within personalized specific right and number of it. The rights, which can be distributed to users are as follows.

- Allow digital contents reading only.
- Allow digital contents reading and storage in the same name.
- Allow digital contents reading and storage in the other name.
- Allow erase and change name as maintenance
- Allow running of process thread.
- Allow hard-copy of the digital contents.
- Allow using of clip-board when digital contents are possible to read and write.

Next, the appropriate manner when the user use the protective resource illegally or getting close to it, is as follow.

- Show out warning message.
- Shutdown an application program that runs the illegal used resource.
- Delete and modify of this thing.
- Shutdown a computer system that is tried illegally.

4 RPLA for Linux Base description

As shown in the picture 4-1, the main working mechanism begins under the relation between kernel mode and user mode. It is said that the overall structure enabling kernel mode to provide best result, there has been main role of kernel module. When a user makes program, one can operate a system calling as like naming subroutine process.

While user process calls a system call, the control process turns user mode into kernel mode. After finishing system call, It returns original status.

The Big difference between user mode and kernel mode is that the code that resides in user process, can access memory and kernel space. In contrast the process that is located in user mode is able to access only itself.

The processes in user space do not interfere each other, They are able to access only by calling system calls.

System calls by doing interrupt procedure, which means actual system calls can only operate by calling interrupt routine begins system calling and saves in arch/i386/kernel/entry. The application is the program that is able to work under various situation in user mode.

Application program is consisted of server and agent, agent exchange many information through communication between server and socket.

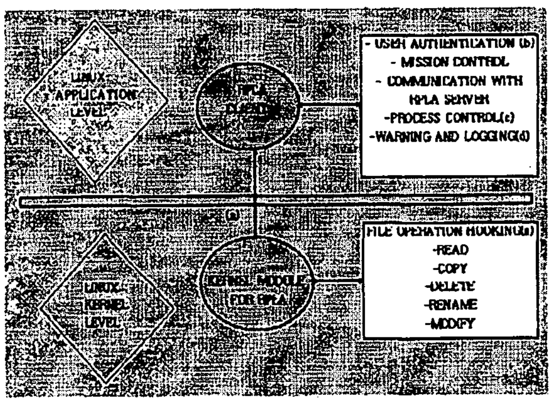
Agent does various works directly to protect information, Server has the ability of user information and allowing level of information protection to provide as a role of donator. In the kernel space, many actions from mouse click and keyboard stroke send to application program.

The file processes such as read(), write(), open(), copy(), delete(), rename(), modify(), chmod(), save(), save as(), close(), etc are receiving from kernel space by hooking mechanism.

As we can get the detail information under the process of agent at the moment.

If the agent detects right information we should protect, the agent gets the message by hooking. Also the

information is the one after comparing in the registered information. If not, Agent restricts application not to use it The mail role of information from kernel module is to cope with DRM solution by doing it. After above procedure, Agent is able to protect information by exchanging dynamically. The main characteristic is to protect knowledge base copyright protection by sensing the illegal actions before or after. The system can operate automatically by exchanging realtime messages between server and agent. This is the right dynamic system to maximize resource management and under various situation.



[The figure 4.1 System construction and function]

5. Consideration

To realize the Linux based agent as above, there are more considerable factors that should be worked out to protect the entire contents protection, besides some necessary development technologies. If these factors are working out, the entire contents protection will be realized safely.

a) Unkillable Process

The user is easy to complete the running process because he has a right about root on his own pc. But it's only possible to protect, not to let the process complete, even if RPLA got a right about root.

b) Hidden Process

Another way to not to be completed the process is that conceal the fact that the process is running.

c) Hidden Proc/

Every process in Linux gets an entry under /proc file system. Therefore, the entry should not let the user know the entry under /proc.

6. Conculsion

As society is getting digitalized, and information-oriented, as digital contents productions are distributed freely, the protection about digital contents copyright authority either law or technology is becoming a serious social issue. As well as Korea, other courtiers are in process to study about improving law system and technical study, like watermarking.

In this paper, it presents **RPLA** System as one method for intellectual property protection and look for the techniques that realize the **RPLA** for Linux, can possibly runs on Linux platform. There are some advantages that can offer a protection function in systematic field, deal with actively, comparing to other digital watermarking technology, and accept variable missions. However, it needs so many caution and effort because Agent runs on not to let the users recognize in PC and should stand on attack by any other propound users who have a lot of knowledge about Linux system. **RPLA** System, in progress of developing and extending now to solve these problems should settle down as a solution, offers a better function about intellectual property protection

7. References

[1] S.Craver, N.Memon, Boon-Lock Yeo and M.Yeung. "Can invisible watermarks resolve rightful ownerships?", Proceedings of IS&P/SPIE Conference on Storage and Retrieval for Image and Video Databases V, San Jose, CA, USA, Feb. 13-14, 1997, vol. 302, pp. 310-321

[2] W.Diffie and M.E.Hellman, "New directions in Cryptography", IEEE Transaction on information theory, Vol. IT-22. NO. 6, November 1976

[3]Korea Information Security Agency http://www.kisa.or.kr

[4] Cox, I.,et al.,"Secure Spread Spectrum Watermarking for Multi media", Technical report, NEC Research Institute, 1995

[5] Stefan Katzenbeisser,Fabien A.P. Petitcolas "Information hiding- techniques for steganography and digital watermarking" Computer Security Series page 149-174

[6] Welsh, Dalheimer, and Kaufman, Running Linux 3rd Edition, O'Reilly

[7] http://www.netcraft.com

[8] M Beck,H Bohme,M Dziadzka,U Kunitz,R Magnus,and D verwomer, Linux Kernel Internals 2nd Edition, Addison Wesley

[9] Alessandro Rubini, Linux Device Drivers, O'Reilly

[10] Richard Stones, Neil Matthew, Beginning Linux Programming, Wrox

[11] Linux IDS Project, http://www.lids.org/

[12]Korean Linux Documentation project -- http//kldp.org

A Study on the System Call for the Protection of Intellectual Property Rights on Linux Base

Heun Kim, Dae-Joon Hwang
The School of Electrical and Computer Science, Sung Kyun Kwan University,
300 Chunchun-Dong, Jangan-Ku, Suwon 440-746, South Korea
heunyong@mail.skku.ac.kr

Abstract

The 21st century we currently live in is the age based on knowledge and information. It is a computer that provides power, which can lead the new wave of information. The digitalization of information is expressed as the digital revolution. Recently, the more important the protection of Intellectual Property Rights(IPR). The variety methods for protection of intellectual property rights are encryption and digital watermarking and access control and so on. In this paper, we propose RPLA system of active tracking mechanism for IPR protection. Also, protective model of intellectual property can use in a protective work of digital contents, adaptive agent and system level protection on client machine.

In conclusion, we describe the implementation of our model in linux system.

Key words : IPR, systemcall, hooking, kernel,RPLA

1. Introduction

In the past, the most important factors have been that of a visible capital and material. However, it has been changing gradually to an invisible knowledge and information from the past.

Linux is easy to modify, and it has been praised for its stability through the verified running systems by many users. It runs on the whole PC, and spread out dramatically because of the low cost installation. In terms of the current trend, it is easy to forecast hereafter that Linux will be the most important system in the major server market. Now is the era of Internet. we need to put on illegal software reproduction on cyberspace, becoming a serious social issue as enlarging Internet market.

As above, these actions must be prohibited. It is natural to make a new rule, to open information to the public, to authorize the right and to protect the creator who provides information.

In this paper, it will introduce our suggested model, which is differentiated with the existing intellectual property technology, and to provide explanation about that model.

2. Construction and function of RPLA System

Either online or offline, RPLA system is able to protect and track of many sources which are used in computer, prepared for digital distribution of digital contents.

Those protect able sources are passive resources (file, directory, port and so on) and active resources (process, thread) and those source are contain in adaptive agent technology and block of illegal approach and use.

2.1. The total structure of the suggested model.

1) Roles of the sever program
   - The basic role of this is giving lots of missions to the agent.
   - Receiving and collecting the results from the agents who are dispatched to several computers.
   - It can get various statistics and gives new missions that are based on the reported results.
   - Supply Network tool or GUI circumstance for convenience of manager.

2) Roles of the agent.
   - Basically, Agents will be dispatched to computers or sites, which are necessary to monitor.
   - Dispatched agents carry out their missions that is given by severs. Mission means duty and order which has to be protected and monitored for the resources of clients.
   - The agents save the result to server when it is on-line and save to any place when it is off-line.

2.1.1 Mission Control

Mission control plays roles that decide the specific works and orders to be carried out by the agent, and transfer these. The order and work is completed one more union. Practical or impractical of the order will be decided through condition of AND, OR, according to demand of protected clients among the missions. The mission which is executed by the agent, will be all changed according as the agent dispatch to which IP area or which user's computer. The kinds of resources that will be monitored by the agent are divided into the active and passive element as shown in figure 3-1. Digital content, which is a main target to monitor, is classified into File, so this comes under the active resource. Identification is to

0-7695-1414-6/01 $17.00 © 2001 IEEE

distinguish the site and target that the agent will dispatch to and work. Number of count is part that decides the number of using and the right of user will be altered depending on who users are. Group resource is the part of managing that means the resources, which the agent will monitor, and the user who will use the resource to be banded one unit as a group.

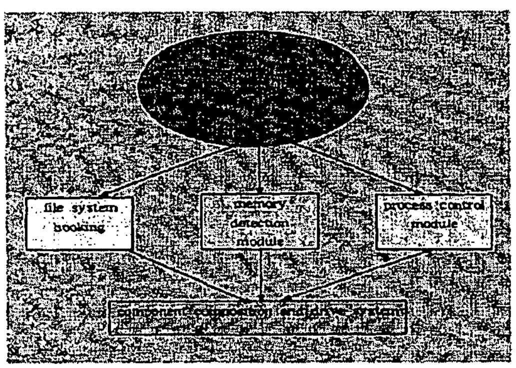
As above, the mission control is that various working and orders which showed in it, are given by consisting of agents which is to suit every computers.

2.1.2 The adaptive agent

The adaptive agent is operating a proper monitor module component to watch and to preserve the resource from the computer, which is dispatched agents, according to the site and an IP address. The agent has to operate downloaded the working component, in the case of receiving the duty for extended works and orders, because the agent is loading the minimum component. And also the adaptive agent technology implies making the appropriate agent to preserve and watch of specific sites and computers.

2.1.3 System hooking

The almost digital contents are file. Such as picture, text, and multimedia, these are all consisted of file. File hooking system enables file to protect and control itself in every hour. The model 2-1 is about RPLA construction module that consisting of file system hooking and the overall contents about file system hooking is as following in chapter 4.



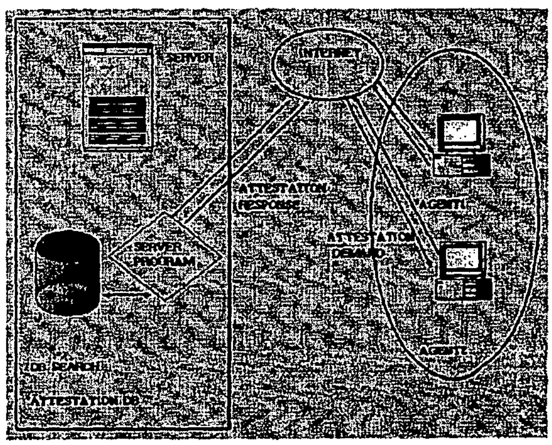
[The figure 2-1. RPLA construction module]

2.1.4 User authorization

This means series of process what the agent distinguish from who is a proper user when the user try to use a resource. The agent has now watched this resource. There are three ways to authorize user from the agent of the suggested model. The standard of classifying these is different depends on the computer is on-line or off-line, and depends on whether the resources which is necessary to keep watch, are many or small. When exchange the data connecting with server in the on-line, use TCP protocol. Firstly, when the computer which is dispatched the agent, is an off-line, it cannot get any related information with the authorization because it cannot connect with sever programs. Therefore, the agent has to have information, which relates with the information of the resources that will watch basically, the information of user who can use those resources, and the right of using number. Secondly, when the number of the resources are a lot, even the computer is an on-line, the agent has to find out the other way if the area is to big to watch.

The ways are as figure 3-6. In the case of used the resources, which is on monitor, the agent requests for an authorization to the server program and that program reply to the requests to the agent, in order to authorize of the resources.

This method minimizes the agent's burden of information storage but it might put stress on network. It may causes problem when the network has an obstacle.



[The figure 2-2. User attestation method for RPLA]

Thirdly, it is a case that number of the protective resources is not that many even though it is on-line. At that time, basically it loads the other information into the agent rather than the protective resources information and a user's information of the resource. It also stores into the agent's database from the server program by downloading the new added resourced information or user's information.

When that happens, it is faster than getting an authorize from connecting by sever each time, and it can be prepared for network area or it's error just like the second case stated above.

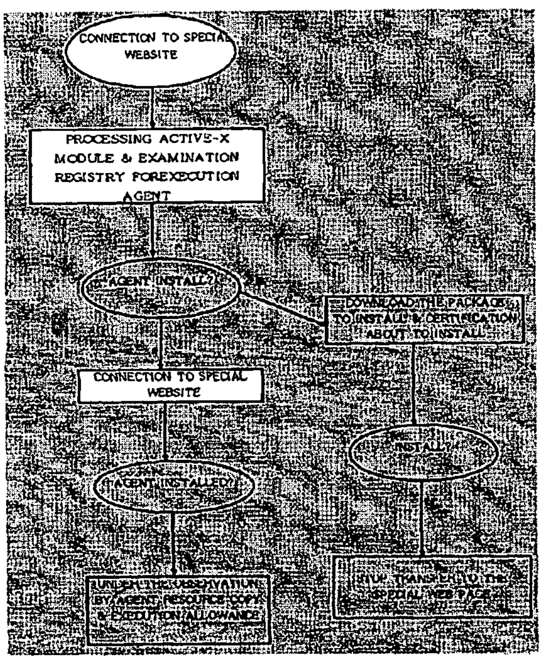
3. How to dispatch the agent.

This method means how to install and run the agent program into the computer system, which wanted to be monitored.

The first, when the user runs portable disk (CD-ROM,

floppy disks) in a computer, the computer installs the program automatically. The agent runs it every time when the computer is booting by user and the user should not be able to notice it is running. In order to hide from the users, the function called stealth is used. If it is used, the agent's running stage does not appear in the process list. As explain about the realization of the stealth's function, it is to resister into service module from running module.

The second, this is a way to dispatch the agent while it is on-line same as the user wants to be downloaded file which is linked by a special web site. Refer to the figure 3. An active component, which is embedded in a web site, runs when it is contacted by specific web site. And it tests out whether the agent's module is running in a connector's computer or not. If not, the package that installs the agent module is downloaded and it runs to install it. If the user avoids installing of it, they cannot open that site. Therefore, the user must install the agent module in that site.



[The figure 3 Agent installation process]

3.1 appropriate manners for decision of the resource's right and an illegal use of the specific resources.

To use easily the protective resources and the management of the user's information who will use that resources. Under a group, many resources and many user's information are registered. The user who is registered in that group is able to use the registered resources within personalized specific right and number of it. The rights, which can be distributed to users are as follows.

- Allow digital contents reading only.
- Allow digital contents reading and storage in the same name.
- Allow digital contents reading and storage in the other name.
- Allow erase and change name as maintenance
- Allow running of process thread.
- Allow hard-copy of the digital contents.
- Allow using of clip-board when digital contents are possible to read and write.

Next, the appropriate manner when the user use the protective resource illegally or getting close to it, is as follow.

- Show out warning message.
- Shutdown an application program that runs the illegal used resource.
- Delete and modify of this thing.
- Shutdown a computer system that is tried illegally.

4 RPLA for Linux Base description

As shown in the picture 4-1, the main working mechanism begins under the relation between kernel mode and user mode. It is said that the overall structure enabling kernel mode to provide best result, there has been main role of kernel module. When a user makes program, one can operate a system calling as like naming subroutine process.

While user process calls a system call, the control process turns user mode into kernel mode. After finishing system call, It returns original status.

The Big difference between user mode and kernel mode is that the code that resides in user process, can access memory and kernel space. In contrast the process that is located in user mode is able to access only itself.

The processes in user space do not interfere each other, They are able to access only by calling system calls.

System calls by doing interrupt procedure, which means actual system calls can only operate by calling interrupt routine begins system calling and saves in arch/i386/kernel/entry. The application is the program that is able to work under various situation in user mode.

Application program is consisted of server and agent, agent exchange many information through communication between server and socket.

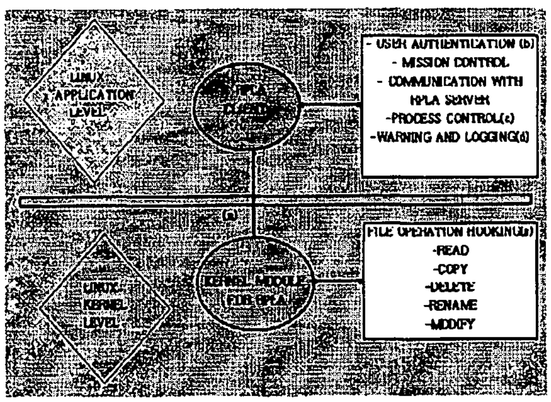
Agent does various works directly to protect information, Server has the ability of user information and allowing level of information protection to provide as a role of donator. In the kernel space, many actions from mouse click and keyboard stroke send to application program.

The file processes such as read(), write(), open(), copy(), delete(), rename(), modify(), chmod(), save(), save as(), close(), etc are receiving from kernel space by hooking mechanism.

As we can get the detail information under the process of agent at the moment.

If the agent detects right information we should protect, the agent gets the message by hooking. Also the

information is the one after comparing in the registered information. If not, Agent restricts application not to use it The mail role of information from kernel module is to cope with DRM solution by doing it. After above procedure, Agent is able to protect information by exchanging dynamically. The main characteristic is to protect knowledge base copyright protection by sensing the illegal actions before or after. The system can operate automatically by exchanging realtime messages between server and agent. This is the right dynamic system to maximize resource management and under various situation.



[The figure 4.1 System construction and function]

5. Consideration

To realize the Linux based agent as above, there are more considerable factors that should be worked out to protect the entire contents protection, besides some necessary development technologies. If these factors are working out, the entire contents protection will be realized safely.

a) Unkillable Process

The user is easy to complete the running process because he has a right about root on his own pc. But it's only possible to protect, not to let the process complete, even if RPLA got a right about root.

b) Hidden Process

Another way to not to be completed the process is that conceal the fact that the process is running.

c) Hidden Proc/

Every process in Linux gets an entry under /proc file system. Therefore, the entry should not let the user know the entry under /proc.

6. Conculsion

As society is getting digitalized, and information-oriented, as digital contents productions are distributed freely, the protection about digital contents copyright authority either law or technology is becoming a serious social issue. As well as Korea, other courtiers are in process to study about improving law system and technical study, like watermarking.

In this paper, it presents **RPLA** System as one method for intellectual property protection and look for the techniques that realize the **RPLA** for Linux, can possibly runs on Linux platform. There are some advantages that can offer a protection function in systematic field, deal with actively, comparing to other digital watermarking technology, and accept variable missions. However, it needs so many caution and effort because Agent runs on not to let the users recognize in PC and should stand on attack by any other propound users who have a lot of knowledge about Linux system. **RPLA** System, in progress of developing and extending now to solve these problems should settle down as a solution, offers a better function about intellectual property protection

7. References

[1] S.Craver, N.Memon, Boon-Lock Yeo and M.Yeung. "Can invisible watermarks resolve rightful ownerships?", Proceedings of IS&P/SPIE Conference on Storage and Retrieval for Image and Video Databases V, San Jose, CA, USA, Feb. 13-14, 1997, vol. 302, pp. 310-321

[2] W.Diffie and M.E.Hellman, "New directions in Cryptography", IEEE Transaction on information theory, Vol. IT-22. NO. 6, November 1976

[3]Korea Information Security Agency http://www.kisa.or.kr

[4] Cox, I.,et al.,"Secure Spread Spectrum Watermarking for Multi media", Technical report, NEC Research Institute, 1995

[5] Stefan Katzenbeisser,Fabien A.P. Petitcolas "Information hiding- techniques for steganography and digital watermarking" Computer Security Series page 149-174

[6] Welsh, Dalheimer, and Kaufman, Running Linux 3rd Edition, O'Reilly

[7] http://www.netcraft.com

[8] M Beck,H Bohme,M Dziadzka,U Kunitz,R Magnus,and D verwomer, Linux Kernel Internals 2nd Edition, Addison Wesley

[9] Alessandro Rubini, Linux Device Drivers, O'Reilly

[10] Richard Stones, Neil Matthew, Beginning Linux Programming, Wrox

[11] Linux IDS Project, http://www.lids.org/

[12]Korean Linux Documentation project -- http//kldp.org